



## Exercícios

1) Analise a complexidade de pior caso dos trechos de código a seguir:

a) Dois *loops* em sequência;

```
for (i = 0; i < N; i++) {  
    sequece of statements;  
}  
  
for (j = 0; j < M; j++) {  
    sequece of statements;  
}
```

b) *Loops* aninhados seguidos por outro *loop*;

```
for (i = 0; i < N; i++) {  
    for (j = 0; j < N; j++) {  
        sequece of statements;  
    }  
}  
  
for (k = 0; k < N; k++) {  
    sequece of statements;  
}
```

c) *Loop* aninhado cuja quantidade de iterações depende do *loop* mais externo;

```
for (i = 0; i < N; i++) {  
    for (j = N; j > i; j--) {  
        sequece of statements;  
    }  
}
```

- 2) Dado um *array*  $a[1 \dots n]$  contendo  $n$  valores, deseja-se construir um novo *array*  $b[i \dots n]$  também formado por  $n$  elementos, onde  $b[i]$  contem a média aritmética dos elementos  $a[0] \dots a[i]$  (para  $i \leq 0 \leq n$ ).

Exemplo:

Entrada:  $a = [3, 1, 2, 6]$

Saída:  $b = [3, 2, 2, 3]$

- 3) Desenvolva um algoritmo de busca capaz de localizar um elemento em um *array* arbitrário (não ordenado) de objetos.
- 4) Deseja-se agora desenvolver um algoritmo de busca capaz de localizar um elemento em um *array* de objetos ordenados.
- 5) Dada uma coleção ordenada de número, desenvolva um programa que identifique (*true/false*) se existe alguma combinação de dois valores da coleção cuja soma seja igual a um resultado preestabelecido.

Exemplo:

$[1, 2, 3, 9]$  soma = 8 (*false*)

$[1, 2, 4, 4]$  soma = 8 (*true*)